Phylogeny-Aware Alignment with PRANK and PAGAN

Löytynoja, Ari

Humana press
2021

Löytynoja , A 2021 , Phylogeny-Aware Alignment with PRANK and PAGAN . in K Katoh (ed.) , Multiple Sequence Alignment . Methods in Molecular Biology , vol. 2231 , Humana press , New York , pp. 17-37 . https://doi.org/10.1007/978-1-0716-1036-7_2

http://hdl.handle.net/10138/337258
https://doi.org/10.1007/978-1-0716-1036-7_2

acceptedVersion

Phylogeny-aware alignment with PRANK and PAGAN

Ari Löytynoja

Abstract

Evolutionary analyses require sequence alignments that correctly represent evolutionary homology. Evolutionary homology and proteins' structural similarity are not the same and sequence alignments generated with methods designed for structural matching can be seriously misleading in comparative and phylogenetic analyses. The phylogeny-aware alignment algorithm implemented in the program prank has been shown to produce good alignments for evolutionary inferences. Unlike other alignment programs, prank makes use of phylogenetic information to distinguish alignment gaps caused by insertions or deletions and, thereafter, handles the two types of events differently. As a by-product of the correct handling of insertions and deletions, prank can provide the inferred ancestral sequences as a part of the output and mark the alignment gaps differently depending on their origin in insertion or deletion events. As the algorithm infers the evolutionary history of the sequences, prank can be sensitive to errors in the guide phylogeny and violations on the underlying assumptions about the origin and patterns of gaps. To mitigate the effects of such model violations, the phylogeny-aware alignment algorithm has been re-implemented in program pagan. By using sequence graphs, pagan can model and accumulate evidence from more complex gap structures than prank does, and incorporate this uncertainty in the inferred ancestral sequences. These issues are discussed in detail below and practical advice is provided for the use of prank and pagan in evolutionary analysis. The two software packages can be downloaded from http://wasabiapp.org/software.

Keywords: phylogeny-aware alignment; evolutionary sequence analysis; insertions and deletions; character homology

1. Introduction

Multiple sequence alignment has a central role in evolutionary sequence analysis, in some studies so central that the alignment and evolutionary inferences should be performed simultaneously

or at least in a tightly coupled manner. The connection between alignment and phylogeny inference was noticed early (1) but the evolutionary consequences of it are still largely ignored by mainstream alignment methods. A probable explanation for this oversight is the historical focus on protein alignments and extensive use of structural benchmarks in the development and comparison of the analysis methods. The use of these benchmarks have produced great alignments for structural studies of proteins but, as noticed by many users of the resulting methods, the very same alignments may be unsuitable for evolutionary analyses.

This chapter focuses on evolutionary sequence alignment and the use of a phylogeny-aware alignment algorithm to infer alignments for evolutionary studies. The de nition of evolutionary homology is central and we will start by discussing that and its correct representation in multiple sequence alignments. We will then introduce { with lots of gures and no equations { the phylogeny-aware alignment algorithm implemented in programsprank and pagan . After detailing the strengths and weaknesses of these methods, we will see what this means in practice and give advice for their use. We will nish with a brief discussion on the future plans for the phylogeny-aware alignment methods.

In the following sections, some methods based on the classical progressive algorithm are criticised and shown to perform poorly. This criticism is based on their performance in evolutionary analyses only and, as demonstrated in other chapters of this book, the alignments they produce can be suitable for other types of analyses. Similarly, the phylogeny-aware algorithm may perform poorly in non-evolutionary alignment tasks and alternative methods should be used.

2. Evolutionary homology in sequence alignment

A multiple sequence alignment represents site-wise homology or \identity" among the characters in di erent sequences. The type of homology/identity denoted by the alignment depends on the application and the intended use of the data: in evolutionary analyses, characters placed in the same alignments columns are believed to be evolutionarily homologous and share a common ancestor; on the other hand, many functional and structural analyses of proteins consider aligned characters positionally identical in proteins' tertiary structures. Evolutionary homology is not the same as structural and positional similarity and the di erence between the two measures is most clearly evidenced by the role of insertions. Two independent insertions at the same position

can lead to similar changes in the structure and the characters inserted independently may thus be considered structurally identical; in contrast, independent insertions { even at exactly the same position { do not share a common ancestor and can never be evolutionarily homologous. To correctly indicate the evolutionary homology of insertions, the characters descending from di erent insertions events should be placed in separate alignment columns (Fig. 1).

If one restricts the analysis to relatively short sequence fragments, one can assume that the sequences evolve by substitutions, insertions and deletions only. The three processes can be assumed to occur at relatively constant (although for di erent processes distinct) rates, substitutions typically being at least an order of magnitude more common than insertions and deletions (2). The three processes di er greatly in their e ect on the sequences and on one's ability infer the events from the data: (i) a character at a certain site can be substituted several times, subsequent substitutions may turn the character state back to an earlier one and characters in di erent evolutionary lineages may independently obtain the same new character state, all still remaining homologous to each other; (ii) an insertion adds new characters to the sequence and subsequent insertions may be nested inside a fragment inserted by a preceding insertion event but, as mentioned above, insertions in di erent lineages are never homologous and evolve independently; and (iii) a deletion removes characters permanently and the characters once deleted cannot be reverted, a potential insertion at the same position introducing new characters that are not homologous with the deleted ones.

By a comparison of two homologous sequences we can detect sites that have undergone substitutions but, without more information, we cannot tell which sequence has changed at which position. In contrast to this, length di erences between two sequences can be explained by deletions of existing characters in one sequence or insertions of new characters in the other. The evolutionary lineage on which the substitution-di erences between two sequences have occurred can be resolved using outgroup sequences and by inferring the character states for the ancestor of the two. Similarly to this, the evolutionary lineages { and thus the types { of insertion/deletion events creating length di erences between two sequences can be inferred using information from phylogenetically related sequences.

Over time, sequences accumulate changes. Despite some di erences in genome sizes, we can assume that sequences tend to retain their approximate length and insertion of new characters is counterbalanced by deletion of others. After a split from a common ancestor, the number of substitution-di erences at homologous positions in descendant sequences increases until the

sequence identity drops to the level expected by random sequences. The e ect of insertions and deletions is very di erent. Assuming that each new insertion is not immediately followed by the deletion of the newly inserted characters, the total number of independent homologous sites within a set of sequence keeps increasing. With more than few sequences in the set, the increase in the number of of independent homologous sites { and thus the number of columns in the alignment representing them { is not signi cantly a ected by deletions as the chances of the same sites being independently deleted in all evolutionary lineages are small. Thus, the total length of the sequence alignment correctly representing the evolutionary homology among the characters is expected to grow roughly linearly with the evolutionary time covered by the di erent sequence lineages. Over long periods of time, the ancestral characters of a neutrally evolving sequence (or sequence region) are expected to be completely replaced by new characters through combinations of insertions and deletions: as a result, the correct evolutionary alignment of highly-diverged descendant sequences should not match a single character. Typically, the more freely-evolving sequence regions are anked by conserved regions (e.g. loops and coils vs. core region in protein sequences) and the alignment is both possible and meaningful.

In practice, the alignment length rarely grows linearly with the evolutionary divergence. If the alignment is performed with methods based on the classical progressive algorithm (3, 4), the alignment length may grow linearly with the number of substitution changes for a while but the growth curves of the two then separate and the alignment length increases only slowly, if at all (Fig. 2). The reason for this is that the classical algorithm does not distinguish insertions from deletions and, inherently, considers all length di erences as deletion events. The use of such biased alignments in evolutionary analysis is likely to lead to erroneous conclusions.

3. Phylogeny-aware alignment

Independent insertions at the same position are not homologous and have to be identi ed to allow for their correct placement in di erent alignment columns. This alone demonstrates that an evolutionarily accurate alignment cannot be generated without considering the phylogeny of the sequences included. In practice, not only are insertions at the same position problematic but the correct alignment of sequences with insertions and deletions at near-by positions requires the identi cation of distinct evolutionary events and their subsequent correct handling.

Progressive alignment algorithms exploit the sequence phylogeny and align the sequences pairwise in the reverse order, starting from the most closely-related ones and, at each step clustering the aligned subsets, progressing towards the root of the tree. A major reason for the use of progressive algorithms is the prohibitive computational complexity of the exact multiple sequence alignment algorithm: with progressive algorithms, the complexity of aligningn sequences of length l is reduced from $O(l^n)$ to $O((n\ \ 1)l^2)$. The additional beauty of the approach is that the algorithm starts with alignments that are expected to be easiest and thus minimises the chances of early alignment errors in its greedy processing of sequences. The classical algorithm does not use the phylogeny for anything else, however, and the placement of gaps { that is, the inference of which characters have been inserted or deleted in the evolutionary past { in the resulting alignments is often phylogenetically implausible (5).

Assuming that the alignment guide tree is correct and that the sequences are relatively closely-related, the progressive alignment approach provides the information necessary to identify insertion and deletion events. The phylogeny-aware progressive algorithm implemented in programs prank (5, 6) and pagan (7) uses outgroup information from the next alignment step to decide if the length di erence observed between the aligned sequences (representing either true extant sequences or internal nodes representing an aligned subset) was caused by an insertion or a deletion (Fig. 3). By identifying the true evolutionary event, the phylogeny-aware algorithm can handle insertions correctly and avoid penalising the single event multiple times in later stages of the alignment.

The two methods are based on the same concept but they di er signi cantly on how they represent the uncertainty of the underlying cause of an observed length di erenceprank is simpler in its design and ags the sites that contain an alignment gap in the immediately preceding stage of the progressive alignment, allowing for free placement of new gaps at agged positions in the very next round. For an insertion, a new gap is created at exactly the same position and the ags indicating the gap are retained; for a gap caused by a deletion, a better alignment is obtained by matching the sites and the ags are removed (Fig. 3).pagan is a reimplementation of the phylogeny-aware algorithm using sequence graphs. Whereas alignment of two sequences withprank creates an ancestral sequence, alignment of two graphs creates an ancestral graph. This graph incorporates the edges of its descendants and, in the case of an alignment gap, contains two alternative paths across the sites (Fig. 3): the path through the unmatched sites represent a deletion in one descendant whereas the path skipping over the sites

represents an insertion in the other descendant. The subsequent alignments can use either of these paths and thus resolve whether the length di erence was caused by an insertion or a deletion. In both methods, the algorithm keeps the inserted sites at the later stages of the progressive alignment and the sequences (or graphs) it reconstructs for the internal nodes of the alignment tree may not re ect the true length of the ancestral sequences. Despite that, the identi cation and marking of the insertion events avoids penalising for the same events multiple times and provides a signi cant improvement over the classical algorithm that, in practice, considers all length di erences as deletions.

Penalisation of a single event multiple times seems an insigni cant error if the procedure nevertheless reconstructs the correct alignment. In trivial alignment tasks that may be the case but in more complex ones the classical algorithm will allow for the matching of insertions with non-homologous characters, the resulting alignments indicating false homologies (Fig. 4). The heuristics proposed to correct for insertion events by lowering the gap cost at sites already containing gaps (e.g. (8, 9)) cannot prevent this; in contrast, they typically cause further errors by moving gaps caused by deletion events at near-by sites to the same columns and produce block-like alignments with alternating gappy and conserved regions (see Fig. 1). The basic version of the phylogeny-aware algorithm greatly reduces the problem but even that cannot completely avoid the matching of independent insertions, especially in the alignment of large datasets in which the chances of mutation events at near-by positions is signi cant (Fig. 2).

As discussed above, the phylogeny-aware algorithm identi es the type of insertion-deletion event and then handles the event accordingly, either creating a new gap or removing the ags indicating the gap. A variant of the algorithm, known as prank $^{+F}$, uses this information to mark sites at which the agged gap is re-used as permanent insertions that cannot be matched at the later stages of the progressive alignment; to prevent overlapping deletions from con rming embedded insertions, the re-use of a gap has to be done for its full length with matching characters at the anking sites. This approach can separate multiple insertions at the same position to independent events without e ect on the placement of gaps caused by deletions (Fig. 4). When the order of aligning the sequences is correct and the sequence sampling is dense enough to call near-by gaps as separate events, prank $^{+F}$ works very well and can reconstruct alignments with lengths very close to the true length (Fig. 2). When the underlying assumptions hold, the method in principle scales up to any number of sequences.

The phylogeny-aware algorithm reconstructs ancestral sequences with information about sites that are believed to be insertions. The ancestral sequences are required for the alignment but they can be useful otherwise, too:prank allows for outputting inferred ancestral sequences, using gaps to indicate sites that are believed to have been later inserted and not present in the ancestors, along with the alignment of the extant sequences. Such alignments are unique and enable studying the process of change and timing certain events to speci c evolutionary branches. In addition to ancestral sequences, the algorithm also infers the type of mutation events that have caused the length di erences between the sequences and can provide this information in the output. Although an experienced user may distinguish insertions and deletions from the gap patterns they create, the marking of gaps caused by insertions and deletions with di erent symbols, as can be done withprank , is helpful.

The explanation and illustration of the agging approach used by prank is slightly simpli- ed and only considers one level of agging. In practice, the algorithm marks the gaps in the immediately preceding alignments and, for the sites not cleared of ags, for the one before that. This procedure prevents long deletions in one branch from masking overlapping insertions in the descendants of its sister branch. For details, see (5, 6).

4. Limitations of the phylogeny-aware algorithm in PRANK

Unlike typical progressive alignment algorithms, the phylogeny-aware algorithm does not align sub-alignments to each other but reconstructs ancestral sequences to represent the parents of sets of aligned descendant sequences and then aligns pairwise these ancestral sequences. Accurate representation of the ancestral sequences, including the detection of inserted and deleted sites, is required for the correct distinction between insertion and deletion events in the subsequent stages of alignment. Correct reconstruction of sequences naturally requires that such ancestral sequences really existed and were true ancestors for the given sets of descendant sequences.

As the ancestral sequences are reconstructed for the internal nodes of the alignment phylogeny, it is crucial that the phylogeny accurately re ects the evolutionary history of the sequences. The role of alignment phylogeny is especially central in the calling of permanent insertions (prank $^{+F}$) that considers the re-use of a agged gap as a con rmation that the gap has been created by an insertion. With the wrong order of aligning the sequences, a deletion may appear as an insertion

and, by marking sites incorrectly as a permanent insertion, the algorithm has to place characters truly homologous to that to separate columns (Fig. 5). Although the resulting alignment is too long and gappy, small errors in the alignment order may not be too serious in typical evolutionary analyses: an incorrect alignment such as that in Fig. 5 does not indicate all true homologies but neither does it contain false homology statements.

In addition to the wrong alignment order, missing data can cause errors with theprank $^{+F}$ variant. The algorithm assumes that alignment gaps are caused by insertions and deletions and then chooses the most plausible explanation of the two. One isolated gap caused by missing data may not be serious but if several sequences lack data at the same region, the gap pattern created may look like an insertion in the complete sequences; when this region is falsely marked as a permanent insertion, the subsequent alignment must place the a ected region in separate columns. As sequences are often truncated at their ends, the marking of terminal gaps as permanent insertions is by default disabled byprank . Similar heuristics unfortunately cannot be provided for missing data in other parts of the sequences.

The phylogeny-aware alignment algorithm assumes that each alignment gap is caused by one insertion or deletion event and that the very next alignment provides information to distinguish between the two types of events. When the sequences are relatively closely related (and, as stated previously, the alignment order is correct), these assumptions are typically valid. If the sequences are more diverged, the chances of independent insertion and deletions events at nearby positions in the adjacent evolutionary branches become signi cant. As a result of this, either the gap created in the rst alignment may be a combination of two or more separate events, or the subsequent alignment of an outgroup sequence fails to con rm the event as an insertion or a deletion due to an overlapping independent event in the neighbouring branch (Fig. 6).

Some of the limitations of the approach and the measures to overcome them can be contradicting. Accurate calling of insertion and deletion events requires densely-sampled sequence sets but the inference of alignment phylogeny for a large dataset is prone to errors (10) and the alignment may therefore su er. Furthermore, incomplete lineage sorting is more likely among closely-related sequences and possibly no single phylogeny correctly re ects the evolutionary history of all sites of a very densely-sampled sequence set.

5. Phylogeny-aware alignment of sequence graphs with PAGAN

The idea of using partial order graphs for sequence alignment is old (11) but its implementation in pagan is novel. In pagan, the graph nodes represent sequence sites, for which the ancestral states are reconstructed using parsimony, while the graph edges indicate the possible paths for the alignment. A crucial feature of pagan's graph edges are the unequal weights that depend on the usage of the edges in the previous steps of the alignment. If certain edges are repeatedly not used, the sites (i.e. graph nodes) that they lead through are likely to be an insertion in one of the descendant sequences. As enough evidence for this is gathered (represented by dashed lines in Fig. 3, top), these edges are completely removed and subsequent matching of the sites is disabled. This behaviour is very similar to permanent insertions of prank +F but the implementation in pagan allows for more flexibility: each edge has a weight and the decision to call something an insertion and remove the flanking edges can be based e.g. on the number of alignments or the phylogenetic distance since the edges were last used. Fig. 4 shows greedy pruning of unused edges after only one alignment (similar to option +F in prank) while the alignments in Fig. 6 require evidence from two additional sequences.

pagan's more flexible modelling and usage of phylogenetic information in the calling of insertions and deletions fixes many shortcomings in prank but even it cannot do miracles. pagan is less sensitive to errors in the alignment order and the phylogenetic inconsistency (e.g. due to incomplete lineage sorting) between different gaps than prank +F (Fig. 5). Similar to prank +F (but not necessarily prank), pagan can align independent insertions at the same position into separate columns (cf. Fig. 1). However, both methods require information from which to call the gap type and also pagan does better with large numbers of densely sampled sequences than with few sparsely sampled ones (Fig. 6). pagan was designed to be faster and, instead of likelihood-based reconstruction of ancestral sites of prank, it uses maximum parsimony (cf. Fig. 3). Furthermore, pagan uses pre-computed scoring matrices, calculated for each alignment step from an evolutionary substitution model using the phylogenetic distances between the sequences, and represents each sequence position, even in the case of ambiguity, with one symbol. In the case of DNA, the standard ambiguity code is applied but for protein and codon sequences pagan can only represent ambiguity between two states and, for more complex cases, uses generic wildcards (X and NNN) that match equally well any other character. Although faster, this simplification can have serious downsides and pagan is likely to perform poorly in the alignment

of a small number of highly diverged sequences. It can perform very well, however, when these highly diverged sequences are accompanied by many other sequences and the sequence sampling is both dense and even (12).

Although pagan can nowadays do de novo multiple sequence alignment, it was initially designed { and still best supported { for phylogeny-aware alignment extension. Here, \alignment extension" means addition of new sequences into an existing multiple sequence alignment such that the relative alignment of the original sequences is not changed. pagan does this in the context of the sequence phylogeny by removing a subtree, aligning sequence(s) to that and then grafting the extended subtree back to its original place in the full phylogeny. The new version of the program, pagan2 , can do this in genomic scale and can generate and extend alignments of closely-related sequences that are upto millions of bases long. pagan can also model uncertainties in the input data and has built-in support for the high error rate of homopolymers tracts in pyrosequencing data and for the high insertion-deletion error rate of the latest generation long-read sequencing data.

6. Phylogeny-aware alignment in phylogenetic analyses

prank and pagan are more \phylogenetic" than many alternative sequence aligners and aim to reconstruct the sequence history with accurate ancestral sequences. However, they are not true phylogenetic algorithms in the sense of e.g. BAli-Phy (13), and are based on a progressive heuristic. The term \phylogeny-aware" refers to their use of phylogenetic information to distinguish insertions from deletions and ability to make gap patterns that correctly re ect the guide phylogeny. With all these phylogeny-related words, it may be surprising that the two programs should be used with caution in phylogenetic analyses.

As discussed above, prank and pagan create gap patterns that re ect the phylogenetic locations of insertion and deletion events (c. Fig 1). In order to do that, they need information about the phylogenetic relationships of the sequences and obtain that from the alignment guidetree. It is important to understand the two methods create these phylogenetic gap patterns by the design of the algorithm and force the patterns to match the guidetree even when the guidetree is incorrect and does not re ect the true phylogenetic relationships of the sequences { or the true pattern of gaps. It is unclear how speci c errors created by the incorrect guidetree a ect

subsequent phylogenetic analyses but the errors are likely to be less random { and thus possibly more harmful { than the errors created by \non-phylogeny-aware" methods.

We studied the performance of prank and pagan in phylogenetic analyses and compared them to heuristics that iterate the alignment and phylogeny inference steps (14). We found that when the sequence evolution perfectly matches the guidetree, prank -generated alignments are very accurate and the phylogenetic trees estimated from them can be more accurate than the trees estimated from true alignments (Fig. 7, left). The result may sound positive but is is actually slightly alarming and indicates that the prank -made errors are biased towards the alignment guidetree. We also saw that both prank and pagan improved with denser sequence sampling and produced more correct alignments for 400 sequences than for a subset of 50 sequences; consistent with this, the phylogenetic trees estimated from the large alignments were more accurate (Fig. 7, middle). However, we also found that the alignment accuracy and the phylogenetic accuracy, as measured by the proportion of matching character pairs (15) and topological distance (16), did not correlate: despite their greater alignment error, the SATe-generated alignments produced in average more correct phylogenetic trees than the prank - and pagan -generated alignments (Fig. 7, bottom).

sate is an iterative method that repeats the alignment and tree inference steps, in this case with mafft (17) and ra xml (18), several times. We developed a similar iterative approach, called Canopy, for phylogeny-aware algorithms and studied if the iteration reduces their guidetree-related error and thus improves their alignment and phylogenetic accuracy. This was indeed the case: Canopy (iterating prank and ra xml) produced more accurate alignments than the three other methods (Fig. 7, top; see also Fig. 3 in (14)) and more accurate phylogenetic trees than one round of prank and ra xml . However, the Canopy-generated trees were more accurate than the sate-generated trees only for the large datasets and the trees estimated from the small datasets, despite the datasets being more accurately aligned, were worse than those generated by sate. On the other hand, one round of pagan and ra xml produced results comparable to those of Canopy, suggesting that the new implementation of the phylogeny-aware algorithm is indeed less a ected by small errors in the alignment guide phylogeny.

The take-home messages from this are somewhat mixed. The phylogeny-aware alignment methods can be used in phylogenetic analyses and they can do very well. However, they have large variance and can do very well, rather poorly or anything between; the iterative approach using a classical alignment algorithm seemed more robust in its performance. The positive nding

is that, perfectly consistent with the theoretical background explained above, prank and pagan improve their performance with denser sampling of sequences. A concrete consequence of this exercise is that the latest version of pagan picks the best approaches from each step and first computes an alignment with mafft, then estimates a tree with FastTree (19) and finally performs the phylogeny-aware alignment based on that tree. A similar iterative approach can, of course, be done manually with prank or using Canopy. Finally, the analysis demonstrates the poor correlation between the accuracy of the alignment and the accuracy of the phylogenetic tree inferred from that. In line with this, modern alignment method comparisons are measuring the performance of the alignments in downstream analyses, not the column-wise accuracy of the alignment itself.

7. Practical advice for the use of PRANK

Evolutionary sequence analysis is based entirely on multiple sequence alignment and the accuracy of the downstream analysis depends on the correctness of the underlying alignment. Alignments produced with prank have been shown to provide accurate inferences of selection on protein-coding sequences (20, 21) and of ancestral sequences (12), and perform well in phylogenetic analyses (22), although the last finding is somewhat controversial due to the role of the guide phylogeny in the phylogeny-aware alignment. Despite its good performance in evolutionary analyses, prank is sensitive to violations of the assumptions made by the algorithm and the users of the program should understand the requirements and limitations of the method.

Alignment phylogeny: The phylogeny-aware alignment algorithm uses the alignment guide phylogeny to distinguish insertions from deletions. The algorithm is therefore sensitive to errors in the guide phylogeny, the variant with permanent insertions (prank $^{+F}$) being especially sensitive. Any prank alignment should be performed using an accurate guide phylogeny: if a high-quality phylogeny is available for the sequence set, it should be used instead of the heuristic phylogeny inferred by the program. In phylogenetic analyses iterative approaches similar to Canopy (14) are recommended.

Evolutionary distances: prank uses the branch lengths provided by the guide phylogeny to recompute the substitution and gap scoring for each alignment step. Depending on the expected evolutionary divergence, a region with several dissimilarities may be considered homologous

and matched (distant sequences), or non-homologous and placed in separate columns (close sequences). Although the algorithm is not sensitive to small deviations in the branch lengths provided, a guide phylogeny with accurate distance estimates should be used when available.

Option +F: Given that the alignment guide phylogeny is correct and the sequence sampling is dense, the variant with permanent insertions (prank $^{+F}$) has been shown to outperform the basic algorithm (5). If the alignment guide phylogeny is likely to contain errors or the input sequences are incomplete (i.e. contain missing data), the option +F can be problematic and the resulting alignment should at least be compared to one produced without it.

Reproducibility: Most pairwise alignments have several equally good solutions. In progressive alignment, the choice between these alternative solutions may trigger larger changes in the later stages of the process and lead to very di erent multiple alignments. Most alignment methods are deterministic and always pick the same solution and thus guarantee to produce the same nal alignment. This practice hides the uncertainty in the data and has lead to post-processing methods to recover the hidden variation (23). By default, prank picks randomly one of the alternative solutions and may produce di erent results on independent runs of the very same data. This behaviour may be disabled if reproducibility is required. By default prank iterates the alignment and the tree inference steps (using the neighbour-joining algorithm for the latter) and keeps the solution that has the best parsimony score. Unlike typical parsimony scores, the prank score considers substitutions as well as insertions and deletions.

Sequence alphabet: prank represents sites at ancestral sequences with vectors of conditional likelihoods for the descendant sub-tree given di erent character states at the parent. This requires $O(A^2)$ computations for each cell in the dynamic programming matrix, where A is the size of the character alphabet, and makes the alignment of sequences with a large alphabet relatively slow. For protein-coding sequences, the alignments performed on codon level has been shown to outperform those done on protein sequences (20, 21). Despite its slower computation, the use of codon alignment is recommended whenever possible. In general, protein-coding DNA sequences should not be aligned as DNA without good reason. If codon alignment is found to be too slow, prank provides an option to translate protein-coding DNA sequences to protein, perform the alignment on protein sequences and back-translate the resulting alignment to DNA.

Sequence sampling: Given that the alignment guide phylogeny is correct and the sequence sampling is dense,prank is unbiased and scales up to any number of sequences. Even if the question in hand would not require an alignment of a large number of sequences, the quality of

the resulting alignment is expected to be better when it is performed for many closely-related sequences than for a small number of distantly-related ones. Unneeded sequences can be removed after the alignment without a ecting the statement of homology among the remaining sequences. prank is not suitable for the alignment of highly diverged sequences.

8. Practical advice for the use of PAGAN

pagan was meant to replace prank but this has not really happened. One reason is that some decisions originally taken to speed up the alignment have compromised the program's performance and pagan currently does poorly in the alignment of very distant protein sequences. Furthermore, the work on pagan has led to unpredicted but still very interesting directions and the program has found a niche of its own. As prank and pagan are based on the same concept, their requirements and limitations are very similar.

Alignment phylogeny: pagan is less sensitive to errors in the guidetree than prank . However, the whole concept of phylogeny-aware alignment is based on the usage of information provided by phylogenetically related sequences and as good a guidetree as possible should be used. By default pagan uses FastTree to estimate the guidetree.

Evolutionary distances: pagan recomputes the scoring matrix for each alignment step from an evolutionary substitution model. The scoring matrix is based on the branch length provided by the guidetree and therefore the branch lengths do matter. Similarly to prank , the scoring matrix is rarely critical for the alignment as long as the distances are of correct magnitude.

Edge pruning: pagan has no option +F but it does a similar trick by removing unused edges. The rules for adjusting the edge weights or removing edges are not thoroughly tested and users are advised to experiment with the parameters if they are unhappy with the result produced with the default parameters. pagan can also be used for pileup alignment where sequences are simply added in the order of appearance as if they would be related by a ladder-like tree. With that, the use of option \keep-all-edges" may be useful as it allows reusing edges even when related sequences are not consecutive in the input le.

Sequence alphabet: pagan was designed speed in mind and, although it uses an evolutionary model to compute the scoring matrix, its representation of ancestral sequences is simplistic, especially for amino acids and codons. pagan represents ambiguous nucleotides with the stan-

dard ambiguity code but, for computational reason it cannot do the same for amino acids and codons. This is not serious in the alignment of rather closely-related sequences and for those pagan is really fast: there is some overhead from the computation of the scoring matrices for larger alphabets, but after that the data type makes no di erence and the alignment of DNA and protein sequences is equally fast.

Sequence sampling: As for prank . pagan requires (and can e ciently utilise) densely sampled sequence and is not suitable for the alignment of highly diverged sequences.

9. Future directions

prank has been shown to perform well in benchmarks assessing the suitability of sequence alignments generated with various methods to di erent types of evolutionary analyses (12,20{22). Despite the use of prank in many phylogenetic analyses, both prank and pagan should be used with caution in analyses where the guide phylogeny is unknown prior to alignment. On the other hand, if the problem with the guide phylogeny can be sorted out, the methods are expected to provide superior alignments for evolutionary analyses, often closely approximating alignments produced with computationally much heavier statistical methods (13, 24).

Although an iterative search strategy should help prank to greatly reduce the problems caused by an incorrect start guide phylogeny, iteration does not decrease the greediness of the algorithm nor can it solve the phylogeny for datasets that have no unique phylogeny, e.g. due to incomplete lineage sorting. These were the reasons to start developing pagan and to re-implement the phylogeny-aware algorithm for the alignment sequence graphs (7). By using additional edges to indicate unresolved gaps and then pruning the unused edges after the alignment of related sequences, one can implement an algorithm very similar to that of prank $^{+F}$ (Fig. 4). The advantages of the graph approach are greater, though, and instead of greedily pruning the edges, they can be given weights or probabilities based on the evidence for the di erent mutation types. Such a exible edge-weighting makes pagan far less sensitive to errors in the guide phylogeny or di erent sites evolving under slightly di erent phylogenies while still allowing for correct separation of independent insertions into columns of their own.

As discussed above and evidenced by methods for co- and joint-estimation of alignment and phylogeny, the multiple sequence alignment should always be seen with the associated phylogeny.

Understanding the alignment and drawing right conclusions from it is much easier when the relationships between the sequences are indicated next to the alignment. For methods such as prank and pagan , the phylogeny is also needed to indicate the relative positions of the ancestral sequences and to visualise the changes happening in di erent evolutionary branches. For this, we have developed Wasabi, a browser-based graphical user interface, that integrates these ideas and provides an easy-to-use access tprank , pagan and many other evolutionary analysis methods (25). The Wasabi server is available at http://wasabiapp.org/ and the use ofprank and pagan within the Wasabi environment is described in detail in another chapter of this book. The two software can be downloaded from the program home pages at http://wasabiapp.org/software. Pre-compiled packages ofprank are provided for Linux, OSX and Windows while the latest version of pagan is currently available for Linux only. prank and pagan are also provided as minimal Docker images that can be run on all platforms. Instructions for installation and usage of di erent versions are given on the program home pages.

References

1. Sanko , D. (1975) Minimal mutation trees of sequences. SIAM J Appl Math, pp. 35{42.
2. Ogurtsov, A., Sunyaev, S., and Kondrashov, A. (2004) Indel-based evolutionary distance and mouse-human divergence. Genome Res, 14, 1610{1616.
3. Hogeweg, P. and Hesper, B. (1984) The alignment of sets of sequences and the construction of phyletic trees: an integrated method. J Mol Evol, 20, 175{186.
4. Feng, D. and Doolittle, R. (1987) Progressive sequence alignment as a prerequisite to correct phylogenetic trees. J Mol Evol, 25, 351{360.
5. Loytynoja, A. and Goldman, N. (2008) Phylogeny-aware gap placement prevents errors in sequence alignment and evolutionary analysis. Science, 320, 1632{1635.
6. Loytynoja, A. and Goldman, N. (2005) An algorithm for progressive multiple alignment of sequences with insertions. Proc Natl Acad Sci U S A, 102, 10557{10562.
7. Loytynoja, A., Vilella, A., and Goldman, N. (2012) Accurate extension of multiple sequence alignments using a phylogeny-aware graph algorithm. Bioinformatics, 28, 1684{1691.
8. Larkin, M., Blackshields, G., Brown, N., Chenna, R., McGettigan, P., McWilliam, H., Valentin, F., Wallace, I., Wilm, A., Lopez, R., Thompson, J., Gibson, T., and Higgins, D. (2007) Clustal W and Clustal X version 2.0. Bioinformatics, 23, 2947{2948.
9. Edgar, R. (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Res, 32, 1792{1797.

10. Liu, K., Raghavan, S., Nelesen, S., Linder, C., and Warnow, T. (2009) Rapid and accurate large-scale coestimation of sequence alignments and phylogenetic trees. Science, 324, 1561{1564.
11. Lee, C., Grasso, C., and Sharlow, M. F. (2002) Multiple sequence alignment using partial order graphs. Bioinformatics, 18, 452{464.
12. Vialle, R. A., Tamuri, A. U., and Goldman, N. (2018) Alignment Modulates Ancestral Sequence Reconstruction Accuracy. Mol Biol Evol, 35, 1783{1797.
13. Suchard, M. and Redelings, B. (2006) BAli-Phy: simultaneous Bayesian inference of alignment and phylogeny. Bioinformatics, 22, 2047{2048.
14. Li, C., Medlar, A., and Loytynoja, A. (2016) Co-estimation of Phylogeny-aware Alignment and Phylogenetic Tree. bioRxiv, p. 077503.
15. Blackburne, B. P. and Whelan, S. (2012) Measuring the distance between multiple sequence alignments. Bioinformatics, 28, 495{502.
16. Robinson, D. F. and Foulds, L. R. (1981) Comparison of phylogenetic trees. Math Biosci, 53, 131{147.
17. Katoh, K., Asimenos, G., and Toh, H. Multiple Alignment of DNA Sequences with MAFFT p. 39{64 Humana Press (2009).
18. Stamatakis, A. (2006) RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. Bioinformatics, 22, 2688{2690.
19. Price, M. N., Dehal, P. S., and Arkin, A. P. (2010) FastTree 2{approximately maximum-likelihood trees for large alignments. PLoS One, 5, e9490.
20. Fletcher, W. and Yang, Z. (2010) The e ect of insertions, deletions, and alignment errors on the branch-site test of positive selection. Mol Biol Evol, 27, 2257{2267.
21. Jordan, G. and Goldman, N. (2012) The e ects of alignment error and alignment ltering on the sitewise detection of positive selection. Mol Biol Evol, 29, 1125{1139.
22. Dessimoz, C. and Gil, M. (2010) Phylogenetic assessment of alignments reveals neglected tree signal in gaps. Genome Biol, 11, R37.
23. Landan, G. and Graur, D. (2007) Heads or tails: a simple reliability check for multiple sequence alignments. Mol Biol Evol, 24, 1380{1383.
24. Novak, A., Miklos, I., Lyngs , R., and Hein, J. (2008) StatAlign: an extendable software package for joint Bayesian estimation of alignments and evolutionary trees. Bioinformatics, 24, 2403{2404.
25. Veidenberg, A., Medlar, A., and Loytynoja, A. (2016) Wasabi: An Integrated Platform for Evolutionary Sequence Analysis and Data Visualization. Mol Biol Evol, 33, 1126{1130.

Fig. 1 The gap patterns in a true alignment re ect the underlying phylogeny of the sequences.
Insertions and deletions create gap patterns (boxes in the alignment; numbered at the bottom) that re ect the phylogenetic locations of the events (black and gray bubbles in the tree). If multiple parallel insertions occur at homologous positions (events 5, 6 and 7), inserted columns can be placed in any order without e ect on the homology statement. In the case of more than two parallel insertions, some inserted fragments are disconnected from the rest of the alignment (here, event 6). The phylogenetic location and type of event 2 is uncertain and it could also be a deletion in the sister branch. Whereas the phylogeny-aware algorithm would re-align the sequences correctly, the classical progressive algorithm (here ClustalW (8)) fails to resolve the true insertion and deletion events.

Fig. 2 The length of the alignment is expected to grow linearly with the evolutionary divergence contained within the sequences. One thousand sequences were simulated under a random tree with the maximum root-to-tip distance of 0.1 substitutions per site. Subsets of 10, 25, 50, 100, 250 and 500 sequences as well as the full datasets were re-aligned with ClustalW (8) and prank , and the length of the resulting alignments is plotted as a function of the length of the tree relating the included sequences. As the insertion-deletion process used for simulation is time-dependent and de ned relative to the substitution rate, the correlation between the two values for the true alignment (black line) is perfect. The two variants of the phylogeny-aware function (prank and prank $^{+F}$) produce alignments with lengths close to the true length whereas a method based on the classical progressive alignment algorithm (ClustalW) over-aligns the sequences and the length of the alignment is seriously underestimated. Solid and dashed lines indicate alignments based on the true and estimated guide trees, respectively. The rectangle in the left plot indicates the area shown on the right.

Fig. 3 The phylogeny-aware algorithm distinguishes insertions from deletions and treats them di erently. The trees on the left represent the evolutionary histories of four short sequences undergoing two substitutions and either an insertion (top) or a deletion (bottom). The coloured trees indicate how the alignment of sequences is divided into three pairwise alignments, each creating an ancestral sequence (Z , Y , X) that is placed at the corresponding internal node and then aligned pairwise with the next sequence. The classical alignment algorithm penalises the single insertion three times (indicated with N; and denote match and mismatch, respectively); in contrast, the phylogeny-aware algorithm implemented in prank ags the gapped site after the rst alignment (indicated by above the sequence) and can then open a new gap at the agged position without a further penalty (indicated by y); pagan models sequences with graphs and represents alternative solutions with additional edges, adjusting the edge weights (shown with di erent line types) depending on their usage. For a deletion, the gap needs to be created only once and the phylogeny-aware algorithm either removes the ag indicating the gap after the second alignment or adjusts the weight of the unused edge. All methods produce the correct alignment.

Fig. 4 The phylogeny-aware algorithm can distinguish and correctly align near-by insertion and deletions. The tree on the left represents the evolutionary history of ve short sequences undergoing two insertion and two deletion events. The coloured tree below indicates how the alignment is divided into pairwise alignments of sequences (or sequence graphs). The resulting alignments are shown on the bottom. The classical alignment algorithm considers length di erences as deletions and cannot place independent insertions into separate columns; often it also moves near-by gaps and indicates false homologies, here resulting in substitutions. A variant of the phylogeny-aware algorithm with greedy calling of insertions, known as prank $^{+F}$, considers the re-use of a agged gap as evidence that the gap was created by an insertion. It then changes the ags indicating a pre-existing gap ( ) to ones indicating a permanent insertion ( ) and does not allow matching of these sites at later alignments. This forces the correct placement of independent insertions into separate alignment columns. The same functionality can be obtained with sequence graphs and greedy pruning of unused graph edges; this is not the default behaviour of pagan , however. See Fig. 3 for the notation.

Fig. 5 The phylogeny-aware algorithm can be sensitive to errors in the guide phylogeny. The tree on the left represents the true evolutionary history and the coloured trees below indicate the right and a wrong order of aligning the sequences. The greedy calling of insertions (prank $^{+F}$) marks agged gaps that are re-used (y) as permanent insertions ( ). When the alignment order is correct (left column), the algorithm works perfectly. If A and C are incorrectly aligned rst (middle column), the subsequent alignment of B appears to con rm an insertion in C although the true event is a deletion shared by A and B . As the insertion in column 4 is marked permanent ( ), the site belonging to that columns has to placed in a columns of its own. The resulting alignment is too long and gappy. prank (without $^{+F}$; not shown) and pagan (right column) are not a ected by this error in the guide phylogeny and produce the correct alignment. See Fig. 3 for the notation.

Fig. 6 Correct identi cation of independent insertion and deletion events requires closely-related sequences. The tree on the left represents the true evolutionary history and the coloured trees below indicate dense and sparse sampling of sequences. With dense sampling of sequences (middle tree) each insertion and deletion event can be identi ed using the information from the next alignment and the correct homology is recovered. With sparse sampling (bottom tree), the insertion in A cannot be identi ed because of a deletion at an adjacent position in D . As a result, the independent insertions in A and E are incorrectly matched. prank +F and pagan are similarly a ected by the sequence sampling; although the two methods place the insertions in di erent order, the resulting alignments are e ectively the same.

Fig. 7 PRANK and PAGAN produce better alignments with dense sampling of sequences. Simulated data were aligned with prank , pagan and two iterative approaches (14), and the accuracy of the alignments (top) and the phylogenetic trees inferred from them (bottom) were evaluated. Alignment accuracies of all phylogeny-aware methods improved with denser sampling of sequences while that of sate decreased. Although similar improvements were seen in topological accuracies, the correlation was not perfect and in many cases sate produced the most correct trees. For the very largest datasets, Canopy produced both the most accurate alignment and the most accurate phylogenetic tree, closely followed by pagan . The bottom row indicates the type of guidetree used in prank and pagan alignments; all phylogenetic trees were inferred with ra xml .